# Recurrent Variational Autoencoders for Learning Nonlinear Generative Models in the Presence of Outliers

Yu Wang* Bin Dai† Gang Hua# John Aston* David Wipf#

***Abstract*—This paper explores two useful modifications of the recent variational autoencoder (VAE), a popular deep generative modeling framework that dresses traditional autoencoders with probabilistic attire. The first involves a specially-tailored form of conditioning that allows us to simplify the VAE decoder structure while simultaneously introducing robustness to outliers. In a related vein, a second, complementary alteration is proposed to further build invariance to contaminated or dirty samples via a data augmentation process that amounts to recycling. In brief, to the extent that the VAE is legitimately a representative generative model, then each output from the decoder should closely resemble an authentic sample, which can then be resubmitted as a novel input ad infinitum. Moreover, this can be accomplished via special recurrent connections without the need for additional parameters to be trained. We evaluate these proposals on multiple practical outlier-removal and generative modeling tasks involving nonlinear low-dimensional manifolds, demonstrating considerable improvements over existing algorithms.**

***Index Terms*—Deep Generative Models, Variational Autoencoder, Robust PCA, Outlier Removal, Variational Bayesian Model, Deep Learning .**

## I. INTRODUCTION

AUTOENCODERS can be viewed as nonlinear generalizations of PCA, capable of producing low-dimensional representations of data lying on or near a manifold [1], [2]. The model consists of two parts: an encoder which computes a low-dimensional representation, and a decoder that uses the latent representation to predict the original input. While serving as one of the most widely-used unsupervised learning approaches, autoencoders are not probabilistic generative models, and hence cannot be directly used to estimate new samples from some target distribution.[1] To address this limitation (among other things), the recently popular variational autoencoder (VAE) replaces the deterministic encoder and decoder with parameterized distributions, and fits them to the data using a principled variational bound that can be optimized using stochastic gradient descent [3], [4]. For both model components, when applied to continuous data it is typical to assume Gaussian distributions with means and covariances computed by individual deep networks.

In addition to its role as a tractable deep generative model, we have argued in a companion work [5] that the basic VAE model is sometimes capable of handling large but relatively sparse outliers, at least provided that the decoder covariance is sufficiently complex/deep. This observation represents our launching point herein, where the goal is to explore several modifications of the canonical VAE pipeline that refine its natural ability to digest dirty, or highly corrupted data and produce a viable low-dimensional representation as though the data had been clean to begin with. To this end, we first present detailed background information regarding the basic VAE model in Section II. We then proceed to our contributions as follows.

In Section III we derive a particular form of conditional autoencoder that jettisons the need for explicitly learning a complex decoder covariance model to handle inputs with gross corruptions. In brief, by conditioning on the sample indices themselves in a precise way, we are able to analytically solve for these covariances in terms of other model parameters (without the need for actually training them) leading to a significantly condensed decoder with many nice attributes related to scale invariance and local minima smoothing when removing sparse outliers.

*Y. Wang (yw323@cam.ac.uk) and J. Aston (jada2@cam.ac.uk) are with the Pure Mathematics and Statistic Department, University of Cambridge, UK. Y. Wang and J. Aston are sponsored by the EPSRC Centre for Mathematical Imaging in Healthcare, EP/N014588/1.

†B. Dai (daib13@mails.tsinghua.edu.cn) is with Tsinghua University, Beijing, China

#G. Hua (ganghua@microsoft.com) and D. Wipf (davidwipf@gmail.com) are with Microsoft Research.

Yu and Bin contributed equally to this work.

[1]The standard autoencoder formulation involves deterministic encoder and decoder networks that are trained using a simple data-fitting criteria; there is no stochastic machinery in place for actually modeling unknown probability distributions. Note that although sometimes probabilistic methods are used to find good initializations for the autoencoder weights and help avoid bad local minima [2], this is a completely separate issue.

Nevertheless, any estimation task involving contaminated samples will require a large training set to compensate, the collection and management of which may be untenable. In Section IV we describe a novel prescription for extracting maximum utility from available data by recycling each sample after its passage through the VAE pipeline. The premise here is that, to the extent that the VAE is a truly representative generative model, then each output from the decoder should closely resemble an authentic sample, which can then be resubmitted as a novel input ad infinitum as a form of data augmentation. Training is accomplished by adding special recurrent connections to the conditional VAE described above, but no additional parameters are required.

Finally, we empirically examine the above two VAE modifications via a battery of tests in Section V. Highlights include the ability to remove large outliers from handwritten digits and face data with far greater success than traditional VAE networks. Moreover, generated samples do not display the blurry artifacts commonly associated with the Gaussian decoder model of existing VAE models, a common criticism of this approach. In fact, even when clean training data is applied, our modified decoder model produces crisper samples for reasons we will describe later. A portion of this work has appeared in conference proceedings [6]. However, that conference version contains no proofs, fewer empirical results, and reduced analyses and perspectives.

## II. VAE BACKGROUND DETAILS

Since the original introduction of the VAE, numerous variants have been proposed to address perceived shortcomings and improve performance on generative modeling tasks. For example, significant effort has been directed towards expanding the effective representational power of either decoder or encoder modules [7], [8] or devising specializations for applications of interest [9]. Regardless, in this work we focus on adapting the original/canonical form of the VAE from [3], [4] to robustly handle outliers. Although not addressed herein, we anticipate that these modifications could be successfully inherited by a broader class of VAE-like models.

The basic VAE assumes that there exists a distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$ over some random variable $\boldsymbol{x} \in \mathbb{R}^d$ of interest, where $\boldsymbol{\theta}$ are unknown parameters that must be estimated from samples $\boldsymbol{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^n$ collected for this purpose.[2] The latent variables $\boldsymbol{z} \in \mathbb{R}^\kappa$ with agnostic prior $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$ are assumed to reflect a low-dimensional (i.e., $\kappa \ll d$) representation of $\boldsymbol{x}$ that characterize its elemental structure. Figure 1(a) illustrates this VAE generative process.

[2]We will use a superscript $^{(i)}$ to reference all quantities associated with the $i$-th sample.

For non-trivial models with sufficiently rich parameterizations, the marginalization over $\boldsymbol{z}$ will be intractable and there is no closed-form solution for $\prod_i p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})$, which could otherwise simply be optimized via maximum likelihood. To circumvent this problem, the VAE introduces the upper bound $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{X}) \geq -\sum_i \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})$ on the negative log-likelihood, where

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{X}) \triangleq \tag{1}$$
$$-\sum_i \left\{ \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) + \mathbb{KL}\left[ q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right) || p_{\boldsymbol{\theta}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right) \right] \right\},$$

$q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)$ defines some arbitrary approximating distribution parameterized by $\boldsymbol{\phi}$, and $\mathbb{KL}[\cdot||\cdot]$ denotes the KL divergence between two distributions. The latter is always a non-negative quantity, which ensures that the bound is strict. Minimizing $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{X})$ with respect to $\boldsymbol{\phi}$ optimizes the tightness of the VAE upper bound, while minimization with respect to $\boldsymbol{\theta}$ leads to the optimal data distribution with respect to this bound. As will be discussed later, both sets of parameters can be jointly learned using a form of stochastic gradient descent.

Additionally, $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ can be interpreted as an encoder surrogate that defines a conditional distribution over the latent 'code' $\boldsymbol{z}$, while $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ serves as the complementary decoder model since, given a code $\boldsymbol{z}$ it quantifies the distribution over $\boldsymbol{x}$. Furthermore, if we first draw random samples from $p(\boldsymbol{z})$, then the decoder can also be used to generate new samples of $\boldsymbol{x}$ for an application-specific purpose.

By far the most common distributional assumptions for continuous data are

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z), \quad p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x), \tag{2}$$

where the moments $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$ are functions of $\boldsymbol{x}$, parameterized by $\boldsymbol{\phi}$, while $\boldsymbol{\mu}_x$ and $\boldsymbol{\Sigma}_x$ are functions of $\boldsymbol{z}$, parameterized by $\boldsymbol{\theta}$. Technically speaking then $\boldsymbol{\mu}_z \equiv \boldsymbol{\mu}_z(\boldsymbol{x}; \boldsymbol{\phi})$, $\boldsymbol{\Sigma}_z \equiv \boldsymbol{\Sigma}_z(\boldsymbol{x}; \boldsymbol{\phi})$, $\boldsymbol{\mu}_x \equiv \boldsymbol{\mu}_x(\boldsymbol{z}; \boldsymbol{\theta})$, and $\boldsymbol{\Sigma}_x \equiv \boldsymbol{\Sigma}_x(\boldsymbol{z}; \boldsymbol{\theta})$; however, for simplicity we will often omit one or both of these arguments when the intended meaning is clear from context. Additionally, the high-dimensional covariance matrix $\boldsymbol{\Sigma}_x$ (as well as sometimes $\boldsymbol{\Sigma}_z$) is typically assumed to be diagonal. Finally, the *conditional* VAE [8], [10] represents one relevant alteration of the basic framework from above. Here we assume that our attention is shifted to the conditional distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{y})$, where $\boldsymbol{y}$ reflects some salient observable quantity, such as a category label or state variable. Using analogous reasoning as before, given $\boldsymbol{Y} = \{\boldsymbol{y}^{(i)}\}_{i=1}^n$ the encoder and decoder distributions from the VAE upper bound are then revised via conditioning to $q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\right)$ and $p_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}|\boldsymbol{z}, \boldsymbol{y}^{(i)}\right)$ respectively, and all posterior moments include an additional dependency on $\boldsymbol{y}^{(i)}$.

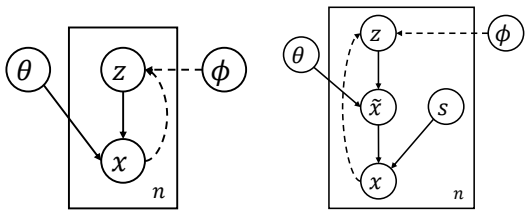



Fig. 1. Graphical model representation. (a): A basic VAE. (b): Our proposed outlier-robust adaptation discussed in Sections III and III-C.

## III. THE iCONDITIONAL VAE AND OUTLIER ARBITRATION

Assuming the decoder covariance $\boldsymbol{\Sigma}_x$ is sufficiently complex, then its diagonal elements can potentially mirror the outlier profile in $\boldsymbol{X}$, with corrupted samples of $\boldsymbol{x}$ producing large values in the corresponding diagonal elements $[\boldsymbol{\Sigma}_x]_{jj}$, and vice versa, clean samples driving $[\boldsymbol{\Sigma}_x]_{jj}$ towards zero, sometimes provably so [5]. To the extent that we believe our data emerge from such a contaminated source, the VAE represents a viable choice for nonlinear, outlier-robust dimensionality reduction or generative modeling. Of course this comes with a significant cost, namely, in practice we must actually train a complex decoder covariance model capable of detecting dirty samples. In this section we describe a convenient workaround based on the conditional VAE.

More specifically, we assume a conditional VAE where the observed latent variables are simply scalars satisfying $y^{(i)} = i$, the sample index itself, a model we refer to as the *iConditional VAE* or iC-VAE. In a broad sense, this conditioning should inject additional representational flexibility into the model since it allows each of the moment functions $\boldsymbol{\mu}_x$, $\boldsymbol{\Sigma}_x$, $\boldsymbol{\mu}_z$, and $\boldsymbol{\Sigma}_z$ to vary in form across each sample. As it turns out however, without loss of generality we may assume that $\boldsymbol{\mu}_z(\boldsymbol{x}, y; \boldsymbol{\phi}) = \boldsymbol{\mu}_z(\boldsymbol{x}; \boldsymbol{\phi})$, and $\boldsymbol{\Sigma}_z(\boldsymbol{x}, y; \boldsymbol{\phi}) = \boldsymbol{\Sigma}_z(\boldsymbol{x}; \boldsymbol{\phi})$, since given a specific sample $\boldsymbol{x}^{(i)}$, the index parameter $y^{(i)} = i$ actually provides no additional information of value, i.e., all subsequent results will ultimately hold with or without this dependency. So this particular conditioning has no impact on the effective encoder, and the KL regularization term is unaffected. We also constrain that $\boldsymbol{\mu}_x(\boldsymbol{z}, y; \boldsymbol{\theta}) = \boldsymbol{\mu}_x(\boldsymbol{z}; \boldsymbol{\theta})$, leaving the decoder mean unchanged (as discussed later in Section III-B, this constraint may be invoked *w.l.o.g.* in certain settings anyway).

In contrast, the proposed conditioning opens a convenient entry point for side-stepping the responsibility of training a huge $\boldsymbol{\Sigma}_x$ via the following downstream effects. First, it is convenient [11] to re-express the conditional VAE upper bound as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{X}) \;&\equiv\; \sum_i \Big( \mathbb{KL}\left[ q_{\boldsymbol{\phi}}\left( \boldsymbol{z} | \boldsymbol{x}^{(i)}, y^{(i)} \right) || p(\boldsymbol{z}) \right] \\ &-\; \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}, y^{(i)})} \left[ \log p_{\boldsymbol{\theta}}\left( \boldsymbol{x}^{(i)} | \boldsymbol{z}, y^{(i)} \right) \right] \Big), \end{aligned} \tag{3}$$

where given the Gaussian assumptions,

$$2\mathbb{KL}\left[ q_{\boldsymbol{\phi}}\left( \boldsymbol{z} | \boldsymbol{x}, y \right) || p(\boldsymbol{z}) \right] \equiv \operatorname{tr}\left[ \boldsymbol{\Sigma}_z \right] + \| \boldsymbol{\mu}_z \|_2^2 - \log |\boldsymbol{\Sigma}_z| \,. \tag{4}$$

Then for a single sample, and given the independence of both $\boldsymbol{\mu}_x$ as well as the encoder from $y^{(i)}$, we have

$$\begin{aligned} &-2\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}, y^{(i)})} \left[ \log p_{\boldsymbol{\theta}}\left( \boldsymbol{x}^{(i)} | \boldsymbol{z}, y^{(i)} \right) \right] \\ &\quad = \int \Big[ \left( \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_x \right)^\top \left( \boldsymbol{\Sigma}_x^{(i)} \right)^{-1} \left( \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_x \right) \\ &\qquad + \;\; \log \left| \boldsymbol{\Sigma}_x^{(i)} \right| \Big] q_{\boldsymbol{\phi}}\left( \boldsymbol{z} | \boldsymbol{x}^{(i)} \right) d\boldsymbol{z}, \end{aligned} \tag{5}$$

where we adopt the notation $\boldsymbol{\Sigma}_x^{(i)} \triangleq \boldsymbol{\Sigma}_x(\boldsymbol{z}, y^{(i)}; \boldsymbol{\theta}) = \boldsymbol{\Sigma}_x(\boldsymbol{z}, i; \boldsymbol{\theta})$. If for each $i$ we can minimize

$$\left( \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_x \right)^\top \left( \boldsymbol{\Sigma}_x^{(i)} \right)^{-1} \left( \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_x \right) + \log \left| \boldsymbol{\Sigma}_x^{(i)} \right| \tag{6}$$

over $\boldsymbol{\Sigma}_x^{(i)}$ independently for all values of $\boldsymbol{z}$, then we will necessarily also minimize (5). Fortunately this is possible if we grant $\boldsymbol{\Sigma}_x^{(i)}$ unlimited capacity to represent any function and knowledge of $i$ as allowed by conditioning. Hence taking derivatives of (6) with respect to $\boldsymbol{\Sigma}_x^{(i)}$, equating to zero and solving, we find that the optimal covariance, when forced to be diagonal (the default assumption used with VAE models as mentioned previously) is given by

$$\operatorname{diag}\left[ \boldsymbol{\Sigma}_x(\boldsymbol{z}, i; \boldsymbol{\theta}) \right] = \left( \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_x \right)^2, \tag{7}$$

where the squaring operator is understood to apply element-wise, and diag[·] converts vector-valued inputs to a diagonal matrix, and square matrix-valued inputs to a vector formed from the diagonal (e.g., as defined in the Matlab computing environment). Plugging this value back into (5) and ignoring constants we find that the overall VAE objective reduces to

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{X}) \equiv \sum_i \Big\{ &\operatorname{tr}\left[ \boldsymbol{\Sigma}_z^{(i)} \right] - \log \left| \boldsymbol{\Sigma}_z^{(i)} \right| + \| \boldsymbol{\mu}_z^{(i)} \|_2^2 \\ &+ \;\; 2 \sum_j \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})} \left[ \log \left| x_j^{(i)} - \mu_{x_j} \right| \right] \Big\}, \end{aligned} \tag{8}$$

where $\boldsymbol{\mu}_z^{(i)} \triangleq \boldsymbol{\mu}_z(\boldsymbol{x}^{(i)}; \boldsymbol{\phi})$ and $\boldsymbol{\Sigma}_z^{(i)} \triangleq \boldsymbol{\Sigma}_z(\boldsymbol{x}^{(i)}; \boldsymbol{\phi})$. Therefore, although a potentially high capacity $\boldsymbol{\Sigma}_x$ in the original VAE model is needed to arrive at something even approximating (8), the net effect of this assumption can lead to a dramatic overall simplification.

Furthermore, from this expression we observe that what was once effectively a quadratic penalty on the errors $x_j^{(i)} - \mu_{x_j}$ is now replaced with a $\log(\cdot)^2$ term, which

as a concave non-decreasing function [12], heavily favors $x_j^{(i)} - \mu_{x_j} \to 0$, while at the same time applying only soft penalization for large values. Such a regularization effect is the cornerstone of sparse estimation algorithms, and hence we may expect that this construction will ultimately be useful for the removal of large yet sparse outliers. Additionally, this regularizer can be viewed as the negative logarithm of the Jeffreys prior on the squared errors, with a number of notable advantages described next.

### A. CHARACTERISTICS OF THE JEFFREYS DISTRIBUTION

As a non-informative prior for quantities such as error variances [13], the Jeffreys distribution $p(e) \propto \frac{1}{e}$ (which as an improper prior does not integrate to one) displays a unique form of scale invariance. In particular, the probability that an error $e = (x - \mu_x)^2$ is between 1 and 10, equals the probability that it is within 10 and $10^2$, or equivalently, between $10^{-2}$ and $10^{-1}$. More generally, the probability that $e$ is within any scaling window is given by $P\left(e \in \left[\eta^k, \eta^{k+1}\right]\right) \propto \log \eta$ for any scale factor $\eta \geq 1$ and any integer $k$ (positive or negative). Therefore outlier arbitration is carried out equally regardless of how any particular data set or network output is scaled.

In contrast, other selections would require special tuning to align with a scale-appropriate range of the distribution. For example, although robust $\ell_p$-norm-based penalties $\sum_j e_j^{p/2}, \quad p \leq 1$ (which can be derived from a generalized Gaussian distribution) also discount large errors/outliers [14], their behavior will be highly dependent on the scale at which outliers are differentiated from inliers, meaning that data in the $[10, 10^2]$ range will be treated very differently than data in the $[10^{-2}, 10^{-1}]$ range.

But there is potential complication associated with the aggregate $\log(\cdot)^2$ penalty arising from the Jeffreys distribution if applied in the context of a traditional autoencoder, the latter of which emerges if we fix $\mathbf{\Sigma}_z = \mathbf{0}$ in the VAE framework and remove the now undefined KL term. Simply put, this penalty will introduce a combinatorial constellation of locally minimizing solutions owing to the infinite regress as any $\left(x_j^{(i)} - \mu_{x_j}\right)^2$ drifts towards zero. In fact, just a single site with $\left(x_j^{(i)} - \mu_{x_j}\right)^2 \approx 0$ can drive the objective towards minus infinity, regardless of the quality of the overall reconstruction at other locations. Hence the energy landscape will be plagued with a combinatorial number of degenerate, infinitely deep extrema.

Fortunately, within the iC-VAE framework, the Jeffreys-based penalty occurs inside of an expectation operator, which smooths over these degenerate pits.[3] However there exists an important exception: if the covariance $\mathbf{\Sigma}_z^{(i)}$ becomes degenerate, e.g., $\mathbf{\Sigma}_z^{(i)} \to \varepsilon \boldsymbol{I}$ with $\varepsilon$ approaching zero, then $q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right) \approx \delta\left(\boldsymbol{\mu}_z^{(i)}\right)$ and

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})}\left[\log\left|x_j^{(i)} - \mu_{x_j}\right|\right] \approx \log\left|x_j^{(i)} - \mu_{x_j}^{(i)}\right|, \quad (9)$$

where $\mu_{x_j}^{(i)} \triangleq \mu_{x_j}\left(\boldsymbol{\mu}_z^{(i)}; \boldsymbol{\theta}\right)$. But the $-\log\left|\mathbf{\Sigma}_z^{(i)}\right|$ term in (8) will normally prevent this from happening since any $\mathbf{\Sigma}_z^{(i)} \to \varepsilon \boldsymbol{I}$ would have a large, counteracting positive contribution. Roughly speaking then, within the VAE framework, the only way we can ever encounter degeneracies introduced by the Jeffreys distribution is if

$$\sum_{j=1}^{d} \mathcal{I}\left[\left(x_j^{(i)} - \mu_{x_j}^{(i)}\right)^2 < \varepsilon\right] > \sum_{k=1}^{\kappa} \mathcal{I}\left[s_k\left(\mathbf{\Sigma}_z^{(i)}\right) < \varepsilon\right] \quad (10)$$

where $\mathcal{I}[\cdot]$ is and indicator function and $s_k(\cdot)$ returns the $k$-th singular value of a matrix.[4] In this situation, the higher dimensionality of the data fit term could outweigh the KL regularizer leading to the collapsed situation under review. But the KL regularization from the VAE framework still provides a valuable service by confining these degeneracies to special cases, and these special cases may be desirable solution points to begin with since they often represent a configuration whereby most data are fit snugly, except for a few exceptions that likely correspond with outlier locations. We will discuss this further in Section IV with a more concrete example. Additionally, further details about how the VAE (and the iC-VAE by inheritance) smooths away bad degenerate solutions, favoring data fit errors exactly aligned with true outlier locations, can be found in [5].

### B. iCONDITIONAL VAE WITH AFFINE DECODER MEAN

After optimizing $\mathbf{\Sigma}_x$ away as described previously, for analysis purposes in this section we consider the case where $\boldsymbol{\mu}_x$ is restricted to be affine, while the encoder moments can have potentially infinite capacity. Although the affine-constrained $\boldsymbol{\mu}_x$ with full conditional plumage would be given by $\boldsymbol{\mu}_x(\boldsymbol{z}, y; \boldsymbol{\theta}) = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{h}y + \boldsymbol{b}$, where $\{\boldsymbol{W}, \boldsymbol{h}, \boldsymbol{b}\} \subset \boldsymbol{\theta}$ represent parameters to learn, it can be shown that in fact the optimal value for $\boldsymbol{h}$ is typically zero. We therefore choose to omit this extra factor consistent with earlier assumptions and ease of presentation.

[3]Note that $\int_{0^+}^{\infty} \log u^2 \cdot p(u) du$ is finite and well-behaved when $p(u)$ is a Gaussian distribution, analogous to the last term in (8).

[4]It is also possible to have trivial degeneracies when other subtle technical conditions occur (e.g., a constant decoder mean function fit to a single sample), but such situations are unlikely to substantially influence practical problems.

Even with the affine assumption however, the expectation in (8) remains intractable, compromising further direct analysis. Fortunately though we can construct a more transparent upper bound that both retains important properties of (8) while simultaneously lending itself to more detailed inquiry.

*Proposition 1:* Assume that $\boldsymbol{\mu}_x(\boldsymbol{z}, y; \boldsymbol{\theta}) = \boldsymbol{\mu}_x(\boldsymbol{z}; \boldsymbol{\theta}) = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{b}$, while $\boldsymbol{\mu}_z(\boldsymbol{x}, y; \boldsymbol{\phi}) = \boldsymbol{\mu}_z(\boldsymbol{x}; \boldsymbol{\phi})$ and $\boldsymbol{\Sigma}_z(\boldsymbol{x}, y; \boldsymbol{\phi}) = \boldsymbol{\Sigma}_z(\boldsymbol{x}; \boldsymbol{\phi})$ are capable via some internal parameter arrangement of representing any function (infinite capacity). Then given $y^{(i)} = i$, a strict upper-bound on the conditional VAE objective from (8) is given by

$$\sum_i h^{(i)}(\boldsymbol{W}, \boldsymbol{b}) \quad \geq \quad \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{X}), \tag{11}$$

where $h^{(i)}(\boldsymbol{W}, \boldsymbol{b}) \triangleq$

$$\inf_{\boldsymbol{\Lambda}^{(i)} \succ \mathbf{0}} \left(\boldsymbol{x}^{(i)} - \boldsymbol{b}\right)^\top \left(\boldsymbol{\Psi}^{(i)}\right)^{-1} \left(\boldsymbol{x}^{(i)} - \boldsymbol{b}\right) + \log \left|\boldsymbol{\Psi}^{(i)}\right|, \tag{12}$$

$\boldsymbol{\Psi}^{(i)} \triangleq \boldsymbol{\Lambda}^{(i)} + \boldsymbol{W}\boldsymbol{W}^\top$, $\boldsymbol{\Lambda}^{(i)} = \text{diag}[\boldsymbol{\lambda}^{(i)}]$, and $\boldsymbol{\lambda}^{(i)} \in \mathbb{R}^d_+$ represents a vector of non-negative variational parameters for each $i$.

There are several important consequences of this result. First, it is not actually required that $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$ have infinite capacity for Proposition 1 to hold. In reality, we only require that much more lenient *stationarity conditions* are satisfied (these emerge from the proof construction; see Appendix A). Secondly, assuming centered data or $\boldsymbol{b} = \mathbf{0}$, then the upper bound from (11) corresponds with a robust PCA model from [15] derived using completely different principles tied to convex analysis and Fenchel duality theory [16]. This model is designed to decompose a data matrix $\boldsymbol{X}$ via $\boldsymbol{X} = \boldsymbol{L} + \boldsymbol{S}$, where $\boldsymbol{L}$ is a low-rank term, reflecting principal subspaces, and $\boldsymbol{S}$ represents sparse errors or outliers, i.e., a matrix with many zero-valued elements and some possibly large corruptions. So we have tied an established probabilistic robust PCA algorithm directly to a specific conditional VAE model, with the latter inheriting any useful properties of the former, which is decidedly more transparent and devoid of intractable integrals. Thirdly, if both

$$\boldsymbol{x}^{(i)} - \boldsymbol{b} \in \text{span}\left[\boldsymbol{\Psi}^{(i)}\right] \quad \text{and} \quad \text{rank}\left[\boldsymbol{\Psi}^{(i)}\right] < d, \tag{13}$$

then $h^{(i)}(\boldsymbol{W}, b)$ will be unbounded from below, since the quadratic term can be held fixed at a finite value while the log-det term is driven to minus infinity. Moreover, because $\sum_i h^{(i)}(\boldsymbol{W}, \boldsymbol{b})$ is an upper bound on both the conditional VAE objective, as well as ultimately $-\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|y)$ by design, this result then implies that infinite negative peaks exist in the original conditional distribution at data points $\boldsymbol{x}^{(i)}$ that can be well-represented by *fewer* than $d$ degrees of freedom. Note that *any* $\boldsymbol{x} \in \mathbb{R}^d$ can be trivially represented using $d$ degrees of freedom. However, degeneracies in the iC-VAE only occur when the degrees of freedom $\kappa$ from the implicit inlier model, combined with the number of sparse errors, i.e., $\|\boldsymbol{\lambda}^{(i)}\|_0 \equiv \text{rank}\left[\boldsymbol{\Psi}^{(i)}\right] - \kappa$, is less than $d$.[5] This will be a desirable degeneracy to the extent that we seek parsimonious data representations, and is unlikely to occur with samples that do not conform to a robust PCA-like model. Please see [5] for more comprehensive analysis of general VAE models and their connection with robust PCA and outlier removal.

### C. Alternative iC-VAE Derivation

In this section, we re-derive the iC-VAE objective (8) from the perspective of sparse Bayesian learning or automatic relevance determination applied to low-rank modeling [15]. We begin with the graphical model from Figure 1(b). We have a latent variable $\boldsymbol{z}$ which determines the 'clean' data $\tilde{\boldsymbol{x}}$, which is then corrupted by some sparse noise $\boldsymbol{s}$, which follows the Jeffreys prior $p(\boldsymbol{s}) \propto \Pi_{j=1}^d \frac{1}{|s_j|}$. Then we obtain the observed data $\boldsymbol{x}$, and $p(\boldsymbol{x})$ can be expressed as

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{s}) p(\boldsymbol{z}, \boldsymbol{s}) d\boldsymbol{z} d\boldsymbol{s} \tag{14}$$

Similarly, we can write an upper bound on the negative log of $p(\boldsymbol{x})$ by replacing the latent $\boldsymbol{z}$ in a conventional VAE with $(\boldsymbol{z}, \boldsymbol{s})$, giving us

$$\begin{aligned}\mathcal{L} =& \mathbb{KL}\left[q(\boldsymbol{z}, \boldsymbol{s}|\boldsymbol{x})||p(\boldsymbol{z}, \boldsymbol{s})\right] \\ &+ \mathbb{E}_{(\boldsymbol{z}, \boldsymbol{s}) \sim q(\boldsymbol{z}, \boldsymbol{s}|\boldsymbol{x})}\left[-\log p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{s})\right].\end{aligned} \tag{15}$$

Note that we omit $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ in the loss function because they serve different roles in this formulation (in fact they are both absorbed into $q$). We define the prior of $\boldsymbol{z}$ as a standardized Gaussian distribution and the approximated posterior of $\boldsymbol{z}$ as a Gaussian distribution determined by the encoder. The prior of $\boldsymbol{s}$ is defined as the Jeffreys prior as mentioned before while the approximated posterior of $\boldsymbol{s}$ is a Dirac-delta function. In summary then we have

$$\begin{aligned} q(\boldsymbol{z}, \boldsymbol{s}|\boldsymbol{x}) &= q(\boldsymbol{z}|\boldsymbol{x}) q(\boldsymbol{s}|\boldsymbol{z}, \boldsymbol{x}) \\ q(\boldsymbol{z}|\boldsymbol{x}) &= \mathcal{N}\left(\boldsymbol{\mu}_z(\boldsymbol{x}), \boldsymbol{\Sigma}_z(\boldsymbol{x})\right) \\ q(\boldsymbol{s}|\boldsymbol{z}, \boldsymbol{x}) &= \delta\left(\boldsymbol{s} - (\boldsymbol{x} - \boldsymbol{\mu}_x(\boldsymbol{z}))\right) \\ p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{s}) &\propto \exp\left(-\frac{1}{2\lambda}||\boldsymbol{x} - \boldsymbol{\mu}_x(\boldsymbol{z}) - \boldsymbol{s}||_{\mathcal{F}}^2\right), \end{aligned} \tag{16}$$

where $\delta(\cdot)$ is a Dirac-delta function and $\lambda$ is a scalar defining the covariance of $p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{s})$. Denote $\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{\mu}_x(\boldsymbol{z})$. Now consider the second term on the r.h.s of (15). It becomes

[5]Here $\|\cdot\|_0$ refers to the $\ell_0$ norm, or a count of the number of nonzero elements.

$$\begin{aligned}
&\mathbb{E}_{(\boldsymbol{z},\boldsymbol{s})\sim q(\boldsymbol{z},\boldsymbol{s}|\boldsymbol{x})}\left[\frac{1}{2\lambda}||\boldsymbol{x}-\boldsymbol{\mu}_x-\boldsymbol{s}||_{\mathcal{F}}^2+C\right]\\
=&\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}\left[\int\delta\left(\boldsymbol{s}-\Delta\boldsymbol{x}\right)\frac{1}{2\lambda}||\boldsymbol{x}-\boldsymbol{\mu}_x-\boldsymbol{s}||_{\mathcal{F}}^2 d\boldsymbol{s}\right]+C\\
=&\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}\left[\frac{1}{2\lambda}||\boldsymbol{x}-\boldsymbol{\mu}_x-(\boldsymbol{x}-\boldsymbol{\mu}_x)||_{\mathcal{F}}^2\right]+C\\
=&C, \qquad (17)
\end{aligned}$$

where $C$ is a constant, hence we can omit this term. The first term of (15) can be decomposed into two parts as

$$\begin{aligned}
&\int q(\boldsymbol{z}|\boldsymbol{x})q(\boldsymbol{s}|\boldsymbol{z},\boldsymbol{x})\log\frac{q(\boldsymbol{z}|\boldsymbol{x})q(\boldsymbol{s}|\boldsymbol{z},\boldsymbol{x})}{p(\boldsymbol{z})p(\boldsymbol{s})}d\boldsymbol{z}d\boldsymbol{s}\\
=&\int q(\boldsymbol{z}|\boldsymbol{x})\log\frac{q(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})}d\boldsymbol{z}\\
&+\int q(\boldsymbol{z}|\boldsymbol{x})q(\boldsymbol{s}|\boldsymbol{z},\boldsymbol{x})\log\frac{q(\boldsymbol{s}|\boldsymbol{z},\boldsymbol{x})}{p(\boldsymbol{s})}d\boldsymbol{z}d\boldsymbol{s}\\
=&\mathbb{KL}\left[q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})\right]\\
&+\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}\left[\int q(\boldsymbol{s}|\boldsymbol{z},\boldsymbol{x})\log\frac{q(\boldsymbol{s}|\boldsymbol{z},\boldsymbol{x})}{p(\boldsymbol{s})}d\boldsymbol{s}\right]. \qquad (18)
\end{aligned}$$

The first term is exactly the same as the KL term in the conventional VAE. Plugging $q(\boldsymbol{s}|\boldsymbol{z},\boldsymbol{x})=\delta\left(\boldsymbol{s}-\Delta\boldsymbol{x}\right)$ into the second term gives

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}\left[\int q(\boldsymbol{s}|\boldsymbol{z},\boldsymbol{x})\log\frac{q(\boldsymbol{s}|\boldsymbol{z},\boldsymbol{x})}{p(\boldsymbol{s})}d\boldsymbol{s}\right]\\
=&\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}\Big[\int\delta\left(\boldsymbol{s}-\Delta\boldsymbol{x}\right)\log\delta\left(\boldsymbol{s}-\Delta\boldsymbol{x}\right)d\boldsymbol{s}\\
&-\int\delta\left(\boldsymbol{s}-\Delta\boldsymbol{x}\right)\log p(\boldsymbol{s})d\boldsymbol{s}\Big]\\
=&C-\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p(\boldsymbol{s}=\boldsymbol{x}-\boldsymbol{\mu}_x)\right]\\
=&C+\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}\left[\sum_j\log\left|x_j-\mu_{x_j}\right|\right]. \qquad (19)
\end{aligned}$$

It should be noticed that though $\int\delta\left(\boldsymbol{s}-\Delta\boldsymbol{x}\right)\log\delta\left(\boldsymbol{s}-\Delta\boldsymbol{x}\right)d\boldsymbol{s}$ is infinite, it is not related to the parameters thus it can be treated as a constant. Combining these results, the lower bound of the negative log likelihood can be finally be expressed as

$$\begin{aligned}
\mathcal{L} \;=\; &\mathbb{KL}[q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})]\\
&+\sum_j\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}\left[\log\left|x_j-\mu_{x_j}\right|\right]+C, \qquad (20)
\end{aligned}$$

which is equivalent to (8) once we account for multiple samples $i=1,\ldots,n$. This formulation explicitly elucidates the crucial role that outliers play in the iC-VAE development.

## IV. RECYCLING DIRTY DATA BY ADDING RECURRENT CONNECTIONS

Although the iC-VAE model on its own has merits in dealing with contaminated data, there is no substitute for a rich set of training samples if any clean low-dimensional representation is ultimately to be found. In this section we describe a simple, practical procedure for creating additional, virtual samples by recycling the VAE output via recurrent connections. The initial intuition here is straightforward: *if the VAE has accurately captured the true generative process, then output samples should be indistinguishable from input samples*, or at least a subset of input samples correlated with the initial seed sample. And if this is indeed the case, then outputs repeatedly fed back through the VAE encoder and decoder networks should produce a sequence of valid samples. In contrast, divergence of this sequence would suggest the accumulation of significant deviations from the true generative process.

Overall, this recurrent structure serves as a form of automatic data augmentation. The network has technically "seen" a wider range of training data, since each partially corrected sample, or perturbed inlier sample, can be viewed as a new input containing attributes not found in the original training data. This includes samples where only a portion of the outliers have been removed, implying that the network will be forced to deal with a much larger breadth of corrupted support patterns. And crucially, the iC-VAE objective is applied to each recurrent loop, leading to an overall process we refer to as a *recurrent* iC-VAE or RiC-VAE.

### A. BASIC MODEL DETAILS

To begin, although the integrals embedded in the iC-VAE cost $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\boldsymbol{X})$ cannot be computed in closed form, the simple stochastic approximation

$$\mathbb{E}_{q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)}\left[\log\left|x_j^{(i)}-\mu_{x_j}\right|\right]\approx\log\left|x_j^{(i)}-\mu_{x_j}\left(\boldsymbol{z}^{(i)}\right)\right| \qquad (21)$$

has been shown to be a suitable, unbiased substitute [3], [4] for the original VAE, where $\boldsymbol{z}^{(i)}$ is a sample drawn from $q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)$. Using a reparameterization trick, every $\boldsymbol{z}^{(i)}$ can be constructed such that gradients with respect to $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$ can be propagated through the righthand side of (21). This involves drawing a sample $\boldsymbol{\epsilon}^{(i)}$ from $\mathcal{N}(\boldsymbol{\epsilon};\boldsymbol{0},\boldsymbol{I})$ and then computing $\boldsymbol{z}^{(i)}=\boldsymbol{\mu}_z^{(i)}+\left(\boldsymbol{\Sigma}_z^{(i)}\right)^{\frac{1}{2}}\boldsymbol{\epsilon}^{(i)}$. See [3], [4] for more details.

To avoid later confusion, we now redefine our original data as $\boldsymbol{X}_1=\left\{\boldsymbol{x}_1^{(i)}\right\}_{i=1}^n\equiv\boldsymbol{X}$, where the context of the new subscript '1' will soon become apparent. Likewise we adopt $\boldsymbol{z}_1^{(i)}\equiv\boldsymbol{z}^{(i)}$ for the latent samples described above. Given a specific $\boldsymbol{x}_1^{(i)}$, the basic iC-VAE model will compute the posterior mean $\boldsymbol{\mu}_{x_1}^{(i)}=\boldsymbol{\mu}_x\left(\boldsymbol{z}_1^{(i)}\right)$ via one pass through the network structure. Moreover, by applying (7) we can extract the companion covariance $\text{diag}\left[\boldsymbol{\Sigma}_{x_1}^{(i)}\right]=\left(\boldsymbol{x}_1^{(i)}-\boldsymbol{\mu}_{x_1}^{(i)}\right)^2$ at this same point. From

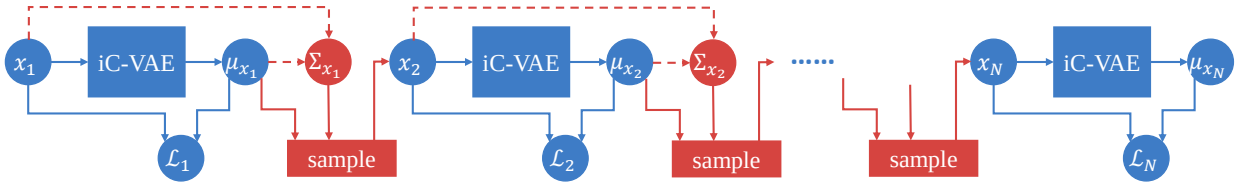



Fig. 2. Structure flow of the RiC-VAE network (sample indices $^{(i)}$ are omitted for simplicity). Solid lines indicate paths in which gradients are backpropagated during training, and $\mathcal{L}_k$ indicates the penalty $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{X}_k)$. Initial data sample $\boldsymbol{x}_1$ passes through the iC-VAE and produces $\boldsymbol{\mu}_{x_1}$. Using (7) the attendant diagonal covariance $\boldsymbol{\Sigma}_{x_1}$ is also computed. A new $\boldsymbol{x}_2$ is then drawn from $\mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}_{x_1}, \boldsymbol{\Sigma}_{x_1}\right)$ and the process repeats. More details in the Appendix B

these two moments, we may then draw a new sample $\boldsymbol{x}_2^{(i)}$ from $\mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}_{x_1}^{(i)}, \boldsymbol{\Sigma}_{x_1}^{(i)}\right)$. Continuing this process across all $i = 1, \ldots, n$, we obtain a new dataset $\boldsymbol{X}_2$.

This operation can be repeated $N$ times, effectively producing a set $\widetilde{\boldsymbol{X}} \triangleq \{\boldsymbol{X}_k\}_{k=1}^N$ of separate datasets, with $N$ times the total number of samples eventually being seen by the network, albeit $N-1$ of these are recycled virtual samples. Nonetheless, these datasets can be used simultaneously during training via the process defined in Figure 2. Importantly, after each pass we include the same iC-VAE objective function applied to the respective recycled data $\boldsymbol{X}_k$, which acts as a form of deep supervision [17], giving the overall RiC-VAE cost

$$\mathcal{L}_N(\boldsymbol{\theta}, \boldsymbol{\phi}; \widetilde{\boldsymbol{X}}) \triangleq \sum_{k=1}^{N} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{X}_k). \tag{22}$$

By penalizing (22), all the iC-VAE units in Figure 2 are effectively forced to shared the same $\boldsymbol{\theta}, \boldsymbol{\phi}$, and hence the overall number of parameters remains unaltered. In terms of training complexity, propagating gradients through the RiC-VAE model is linear in the number of virtual samples $N$. So the required computation is essentially no different than training a regular VAE with $N \times n$ data points, and an $N$ times larger batch size (one actual batch plus $N-1$ virtual batches). Later in Section V we will demonstrate that even $N = 2$ can lead to considerably improved performance, so a dramatically increased training cost need not be a major concern.

### *B. CONNECTIONS WITH ITERATIVE REWEIGHTED COMPRESSIVE SENSING ALGORITHMS*

In the spirit of learning-to-learn [18], learning-to-optimize [19], and other recent attempts to replace or augment conventional iterative algorithms with deep networks estimated from training data [20], [21], [22], the proposed RiC-VAE framework can be viewed as an unfolded iterative algorithm with many trainable parameters. At a high level this follows because there is a now well-recognized link between activations passing through a recurrent neural network such as RiC-VAE, and the iterations of conventional algorithms, which are often structured as $\boldsymbol{x}_{k+1} = \pi(\boldsymbol{x}_k)$, where $\pi$ is some function of signal estimate $\boldsymbol{x}_k$ defined at iteration $k$.

As an illustrative example, consider the family of iterative reweighted $\ell_1$ norm minimization algorithms (IR-$\ell_1$) recently developed for sparse estimation and compressive sensing [23]. Here the objective is to minimize some function $f(\boldsymbol{x})$ that reflects a structured regression task, often of the form

$$f(\boldsymbol{x}) = \|\boldsymbol{u} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \rho\left(\boldsymbol{x}\right), \tag{23}$$

where $\boldsymbol{A}$ is a matrix of feature vectors and, $\boldsymbol{u}$ is an observated vector we would like to represent. In this context, $\rho$ is a penalty that favors some type of structured representation, for instance, pushing most elements of $\boldsymbol{x}$ to zero. However, because the most effective penalties are non-convex, minimizing $f$ cannot be accomplished with typical convex solvers. Instead, the problem is broken down into more manageable convex subproblems and solved iteratively. For example, given the representation $\boldsymbol{x}_k$, the IR-$\ell_1$ algorithm proceeds to iteration $k+1$ via two steps:

$$\begin{aligned} \boldsymbol{z}_{k+1} &\leftarrow g\left(\boldsymbol{x}_k; \boldsymbol{A}\right), \\ \boldsymbol{x}_{k+1} &\leftarrow h\left(\boldsymbol{z}_{k+1}; \boldsymbol{A}, \boldsymbol{u}\right) \\ &\triangleq \arg\min_{\boldsymbol{x}} \|\boldsymbol{u} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \boldsymbol{z}_{k+1}^\top |\boldsymbol{x}|, \end{aligned} \tag{24}$$

where the absolute value operator $|\cdot|$ is understood to apply elementwise, and each iteration is guaranteed to reduce or leave unchanged $f(\boldsymbol{x})$ provided the proper choice for $g$ is used.

We now provide a reinterpretation of this iterative approach in the context of the RiC-VAE. First, the function $g$ plays the role of an arbitrary encoder model, sometimes parameterized by $\boldsymbol{A}$ [24], that computes a set of weights (or latent variables) $\boldsymbol{z}$. However, given that $g$ is handcrafted in an application-specific manner, often based on gradients of some heuristically chosen sparsity penalty with no clear guidelines on the optimal choice, we might expect that a learned replacement would afford some benefit. Either way, once computed $\boldsymbol{z}_{t+1}$ is then fed to the function $h$, which is analogous in functionality to a decoder. Moreover, although $h$ is defined variationally in terms of an optimization problem, it has been

shown that comparable weighted $\ell_1$ regressions can be implemented via a DNN-like decoder structure [25]. Therefore again, it is reasonable to consider replacing this inner-loop optimization step, which could be computationally expensive, with a trainable decoder module.

Furthermore, when we interpret $\{\boldsymbol{A}, \boldsymbol{u}\} \equiv \boldsymbol{y}$ as additional observable latent variables, then a single iteration of (24) accurately maps to a form of handcrafted conditional autoencoder archetype, strengthening the overall analogy further. And of course the incentive to iterate this type of process is significant. As we will later observe in Section V, the empirical behavior of our RiC-VAE model subject to multiple recurrent loops mirrors the improvement seen by IR-$\ell_1$ algorithms. In both cases, initial iterations focus on localizing the optimal support pattern of the significant components of $\boldsymbol{x}_k$, while later iterations refine these solutions. So there is no longer any need for a first pass through the VAE to provide a perfect screening, since further iterations can clean up outliers or imperfections missed during the first pass.

## V. EXPERIMENTAL RESULTS

Training data are not always perfect. Instead of selecting clean images manually, our RiC-VAE is able to recycle dirty data into useful samples. In this section, we demonstrate the advantage brought about by this ability applied to generative tasks and outlier removal problems. Throughout we use $N$ to refer to the number of RiC-VAE passes/recurrencies used during *training*, as distinguished from $M$, the number of RiC-VAE passes applied at *test* time, either for generating novel samples from a random seed or else cleaning a newly introduced corrupted data drawn from the same manifold. These need not always be the same given that a learned model can be iterated for any number of passes. We use RiC-VAE($N$,$M$) to describe the generic case, which implies that iC-VAE equals RiC-VAE($N$=1,$M$=1). It also naturally follows that the test-time complexity of any RiC-VAE($N$,$M$) model will simply scale linearly in $M$ relative to a regular VAE, and this value could conceivably be tuned to match any application-specific requirements.

### A. EVALUATION OF iC-VAE BASELINE

**Image-wise outlier removal:** The Google-30 data [26] includes images returned from 30 different search queries, with roughly 500 images collected per concept. Human labelers then determine which of these are relevant, and which are considered as outliers or irrelevant. This data has been recently used to assess various unsupervised outlier detection algorithms, where the labels themselves are only used for evaluation purposes [26], [27]. We adopt a similar experimental design; however, because the number of samples per query is relatively small, we restrict ourselves to a simple affine iC-VAE model. Regardless, the Google-30 data still provides a useful benchmark for evaluating such a baseline upon which the RiC-VAE ultimately depends.

TABLE I
OUTLIER DETECTION ACCURACY ON GOOGLE-30 DATA.

| Method | UOCL | DRAE | iC-VAE |
|---|---|---|---|
| Average F1 scores | 0.826 | 0.849 | **0.874** |

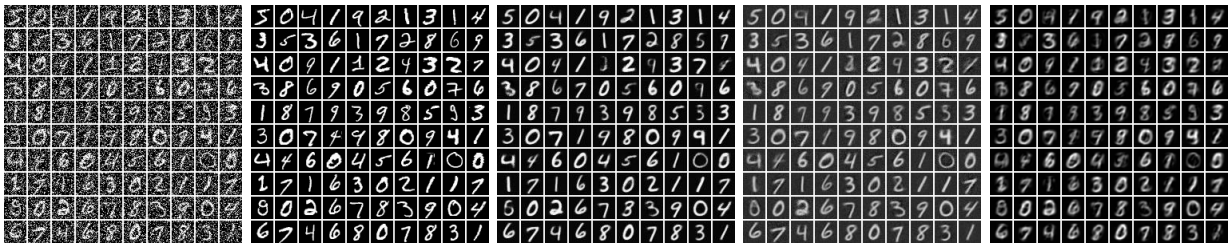
We learn an affine iC-VAE-based model for each search query, and then predict outliers using the thresholding heuristic from applied to residuals from [27]. We also assume that $d = \kappa$, meaning that the iC-VAE must automatically learn any latent low-dimensional structure via its natural regularization process (no tuning of the latent dimension is required). F1 scores from this procedure averaged across all 30 search queries are shown in Table I along side results from two state-of-the-art approaches: a kernel-based max-margin algorithm called UOCL from [26], and an autoencoder-based pipeline DRAE from [27]. Although admittedly these results do not highlight the full flexibility of the RiC-VAE, they nonetheless support the iC-VAE as a viable building block or starting point.

**Pixel-wise noise removal:** We next test the iC-VAE on a standard pixel-wise outlier removal problem, comparing against RPCA and a standard VAE with a learned decoder covariance $\boldsymbol{\Sigma}_x$ (the latter is referred to as LC-VAE). The MNIST handwritten digit data [28] contains 60000 training images of digits, each of size $28 \times 28$. We corrupt $40\%$ of the pixels in each image by randomly replacing the original values with samples drawn uniformly from $[0, 255]$, i.e., salt-and-pepper noise. These 'dirty' training images are then fed to each algorithm with the goal of recovering the original clean MNIST images.

Figures 3(a) and 3(b) present samples of the contaminated images and the corresponding original clean versions respectively. As shown in Figure 3(c), the iC-VAE significantly outperforms its rivals by capitalizing on both its ability to mimic a high-capacity $\Sigma_x$ as well as its flexibility to accommodate a nonlinear manifold (upon with MNIST digits lie). In contrast, Figure 3(d) exposes the disadvantage of using the LC-VAE, where the limited capacity covariance is insufficient, with numerous obvious artifacts in the reconstruction. Finally, the RPCA result in Figure 3(e) demonstrates that a low-dimensional linear subspace inlier model is inadequate for representing MNIST digits.

### B. RiC-VAE OUTLIER REMOVAL PERFORMANCE

**Recovery of clean training data:** The Frey face dataset [4] includes 1965 images, each of size $28 \times 20$.



(a) (b) (c) (d) (e)

Fig. 3. Demonstration of MINIST data denoising performance using iC-VAE, LC-VAE (self-learnt decoder covariance), and RPCA respectively. (a) MNIST training data corrupted with $40\%$ salt-and-pepper noise. (b) Original clean MNIST data. (c) Reconstruction using iC-VAE. (d) LC-VAE (e) RPCA.

We selectively contaminate these images to varying degree using a randomly positioned dark circle mark with random radius. We vary the percentage of training images corrupted in this way, and compare the ability of 8 different models to recover the original, clean face images given only the contaminated source. These include: (a) a convex robust PCA (RPCA) approach from [29] often applied to this problem [30], (b) a conventional autoencoder (AE), (c) an $\ell_2$-VAE, meaning a standard VAE with fixed decoder covariance $\mathbf{\Sigma}_x = \boldsymbol{I}$ as is most commonly assumed [11], (d) a recurrent $\ell_2$-VAE denoted R$\ell_2$-VAE($N$=2,$M$=2), i.e., analogous to RiC-VAE($N$=2,$M$=2) but with fixed decoder covariance, (e) a standard VAE with the a learned decoder covariance called LC-VAE, (f) a recurrent LC-VAE version denoted RLC-VAE($N$=2,$M$=2), (g) an iC-VAE, (h) a RiC-VAE($N$=2,$M$=2). For all VAE models, we use $\kappa = 10$ and a common 3-layer encoder/decoder network structure, with details deferred to the Appendix B. Additionally, all VAE and AE networks share common DNN structures with the exception of different loss layers as stated, and only the VAE has encoder/decoder covariance functions and KL terms. Figure 4 qualitatively illustrates the advantage of the multiple data passes/recycling leveraged by the RiC-VAE at an image corruption level of $60\%$. In fact, even with huge contaminations (e.g., $4^{th}$ and $7^{th}$ columns), the RiC-VAE is still able to reconstruct salient facial details. Moreover, the initial iC-VAE estimate only partially removes the corrupted region, analogous to how initial iterations of IR-$\ell_1$ algorithms only partially recovery the optimal support patterns of sparse representations as discussed in Section IV-B. Complementary quantitative results are presented in Figure 5(a) for all algorithms. Here we observe that traditional methods (i.e., RPCA, AE, $\ell_2$-VAE, LC-VAE) do not produce competitive results, and RLC-VAE exhibits no advantage over LC-VAE since, not surprisingly, learning decoder covariances destabilizes the recycling process. The iC-VAE is adequate at low corruption levels but starts to breaks down above $30\%$.

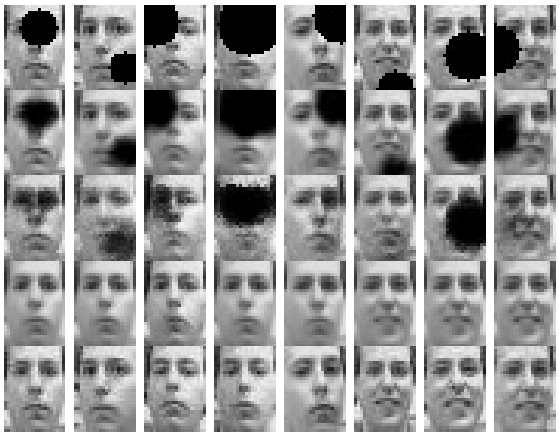
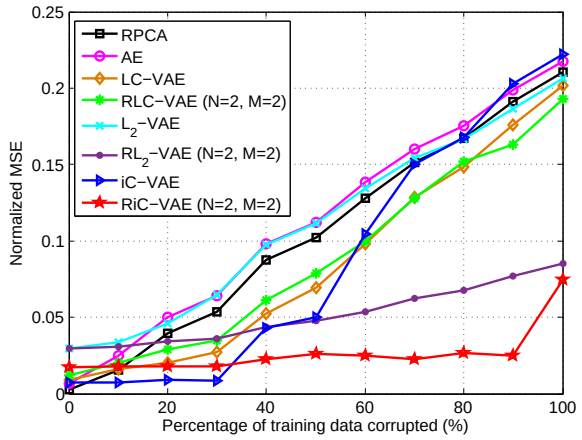


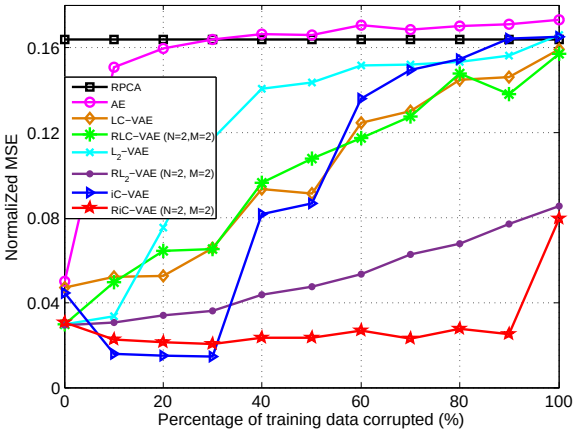
Fig. 4. Visualization of recovery results on Frey face data. 1179 of the 1965 images ($60\%$ ) were corrupted by a randomly positioned dark circle mark with random radius. Each column corresponds to a different database image. *Row 1*: Original contaminated samples. *Row 2*: Reconstructed images using RPCA. *Row 3*: Reconstructed images using an iC-VAE (no recycling). *Row 4*: Reconstructed image from an RiC-VAE($N$=2,$M$=2). *Row 5*: Clean ground truth data without contamination.

In contrast, while the R$\ell_2$-VAE($N$=2,$M$=2) exploits our proposed recycling strategy, without the iC-VAE base network its performance cannot match the RiC-VAE.

**Recovery of a new test set**: We also generated a new test dataset by changing the dirty pattern added to the faces. Specifically, instead of using circle-shaped outliers as applied above, we generate new 'rectangle' dirty patterns having random width, length and location. The 1965 clean Frey face images were corrupted by these new rectangle marks, allowing us to examine the resilience of the previously-trained models to outlier distributions distinct from the original data. Figure 6 demonstrates the dirty pattern in test mode and the corresponding reconstruction results when trained at a $60\%$ corruption level. Figure 5(b) displays the overall reconstruction errors, where the superiority of the RiC-VAE (with $N, M > 1$) is preserved.




(a) Reconstruction MSE on Frey face training data.




(b) Reconstruction MSE on novel test images.

Fig. 5. Evaluation of reconstruction MSE

Fig. 6. Visualizing reconstructions on Frey face test data. *Row 1*: Dirty test data. *Row 2:* Reconstruction by iC-VAE. *Row 3:* Reconstruction by RiC-VAE ($N$=2,$M$=2). *Row 4:* Original clean Frey faces

### *C. RiC-VAE GENERATIVE MODELING PERFORMANCE*

Moving beyond outlier removal, arguably the most common application of VAE models is to the task of generating new samples of $\boldsymbol{x}$ [11]. This section explores RiC-VAE capabilities in this revised context using MNIST handwritten digit data [28]. We first train different models using this data, both clean and dirty versions, and then compare performance on subsequent generative tasks. Models considered include: (a) a standard $\ell_2$-VAE, (b) an iC-VAE, (c) an RiC-VAE($N$=1,$M$=20), (d) an RiC-VAE($N$=5,$M$=1), and (e) an RiC-VAE($N$=5,$M$=20). In all cases $\kappa = 30$, and both encoder and decoder have 3 layers (the Appendix B contains full network structure and training details). By varying $M$, we can examine the quality of generated samples after different passes through the networks at test time.

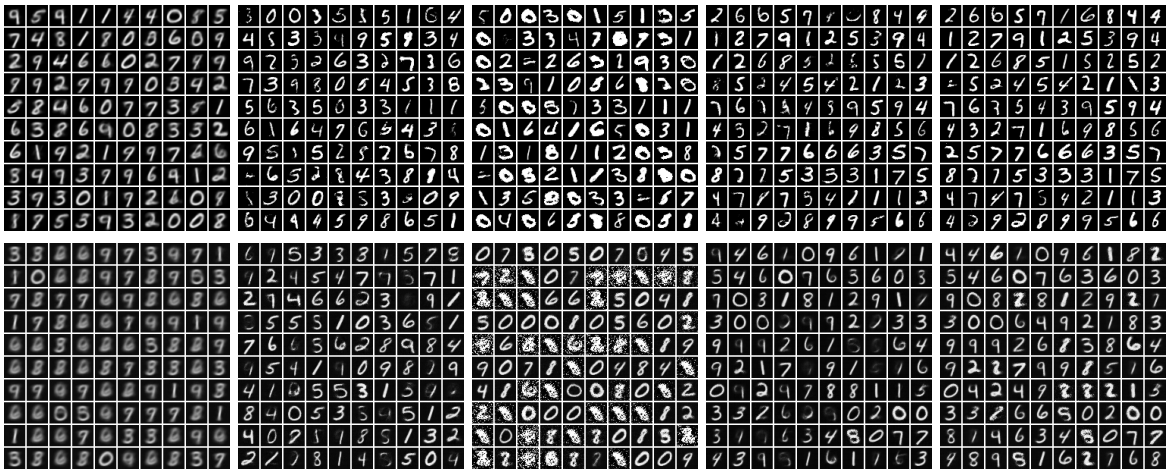
**Results using clean training data:** Here we first use the original MNIST data for training (no corruptions added) and compare the quality of new generated samples obtained by first drawing a latent $\boldsymbol{z}$ from $\mathcal{N}(\boldsymbol{z};\mathbf{0},\boldsymbol{I})$ and then passing the resulting value through the decoder to produce a sample of $\boldsymbol{x}$ [11]. Results from 100 random draws are shown for each method in Figure 7(*top row*). In (a) we observe that the $\ell_2$-VAE produces overly blurry samples, a common criticism, and although the iC-VAE removes this blur in (b), realistic digit shapes are compromised. Next, (c) reveals that cycling through a learned iC-VAE network (i.e., $N$=1) when generating samples introduces new artifacts, since recycling was not used during training, and conversely, in (d) we see that the use of recycling during training has limited value without the attendant recycling at test time generating new samples. Finally, we see that the full RiC-VAE structure produces more authentic digit samples, and that this can be achieved even though the number of training and test passes are not equivalent. Please see Figure 3(b) for original MNIST data examples to compare against.

**Dirty training dataset**: We next repeat the above experiment using corrupted training data; please see Figure 3(a) for visualization of corrupted MNIST samples. Specifically, $40\%$ of pixels are replaced with random values drawn from a uniform distribution over $[0, 255]$, i.e., salt-and-pepper noise. In this more challenging situation, the value of recycling dirty training samples is readily apparent as shown in Figure 7(*bottom row*).

**Statistical validation of generated samples:** If an estimated VAE model truly reflects the underlying latent distributions well, then

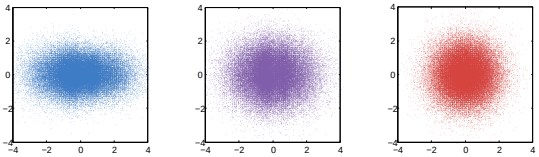
$$\int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\, p_{\boldsymbol{\theta}}(\boldsymbol{x}) d\boldsymbol{x} \approx \tfrac{1}{n}\sum_i q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right) \approx p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z};\mathbf{0},\boldsymbol{I}). \quad (25)$$

To test this hypothesis, we generate samples of $\boldsymbol{z}$ from $\frac{1}{n}\sum_i q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)$ and make scatter-plots of two randomly selected dimensions. Figure 8 shows results for both the iC-VAE and a RiC-VAE with $N$=5 trained on MNIST data; clearly the latter is able to remove some of the heteroscedastic variance of the former. This



(a) (b) (c) (d) (e)

Fig. 7. Samples generated from VAE models trained on MNIST data (please zoom for better viewing). *Top Row*: Results using clean training data. *Bottom Row*: Results using noisy training data. *Columns*: Samples generated from (a) $\ell_2$-VAE, (b) iC-VAE, (c) RiC-VAE($N$=1,$M$=20), (d) RiC-VAE($N$=5,$M$=1), and (e) RiC-VAE($N$=5,$M$=20). For comparison purposes, please see Figure 3(a,b) for examples of both corrupted and clean MNIST samples. In general, the new RiC-VAE($N$=5,$M$=20) samples most closely resemble the clean MNIST data.



(a) (b) (c)

Fig. 8. 2D samples from $\frac{1}{n}\sum_i q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)$ when using (a) iC-VAE, and (b) RiC-VAE with $N$=5. (c) Ideal samples from $\mathcal{N}(\boldsymbol{z};\mathbf{0},\boldsymbol{I})$.

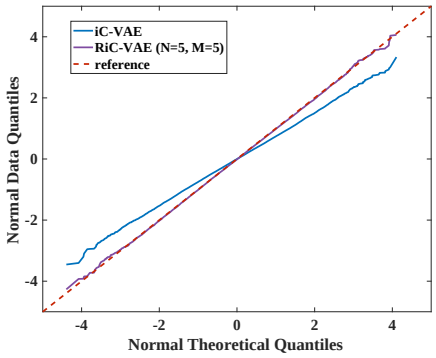
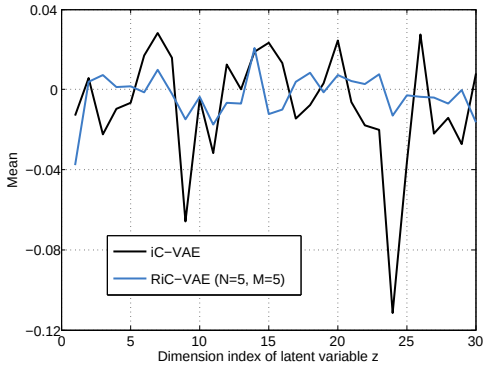
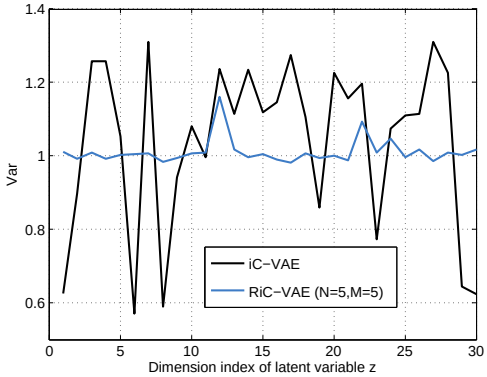



Fig. 9. Q-Q plots of iC-VAE (*blue*) and RiC-VAE (*purple*) models against the reference normal distribution (*dotted red line*). Clearly the RiC-VAE displays a closer fit.




(a) Mean value comparison.




(b) Variance value comparison

Fig. 10. Means and variances of samples drawn from $\frac{1}{n}\sum_i q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)$.

conclusion is further supported via Q-Q plots [31] for the iC-VAE and RiC-VAE against a reference normal distribution as illustrated in Figure 9. Here we observe that, relative to the iC-VAE, quantiles from the RiC-VAE model much more closely align with the reference, corroborating that (25) represents a suitable approximation.

We also consider a higher dimensional comparisons with a standardized Gaussian. In particular, we draw 60000 samples from $\frac{1}{n}\sum_i q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)$ and then compute the mean and variance of each of the $\kappa = 30$ dimensions in $\boldsymbol{z}$ as used with the MNIST data. Ideally these means

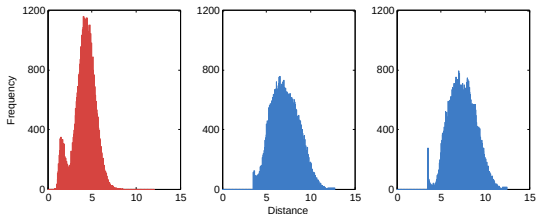



Fig. 11. Evaluation of sample diversity. *Left*: Histogram of nearest neighbor distances in original MNIST data. *Middle*: Histogram of distances between 60000 RiC-VAE($N$=5,$M$=1) samples and their nearest neighbors in MNIST data. *Right*: Same for a RiC-VAE($N$=5,$M$=5).

should be near zero while the variances should be near one. Figure 10 shows the results for an iC-VAE and a RiC-VAE. From this figure it is immediately apparent that the RiC-VAE statistics are far more consistent with the ideal standardized Gaussian (which emerges if the non-Gaussian underlying latent distributions are approximated well) than the iC-VAE, suggesting that our recycling approach indeed better captures the true underlying distributions. Note also that results are similar as $M$ is varied, implying that the statistics of recycled data or multiple passes at test time are stable.

A second important validation issue pertains to sample diversity. In brief, we would like to generate novel samples that are not trivially plagiarized versions of the original training set. To examine this issue, we plot the mean Euclidean distance between each sample generated by a RiC-VAE and its nearest neighbor in the MNIST data. These distances should be as large or larger than the mean distance between each authentic MNIST sample and its nearest neighbor if no copying has occurred. Figure 11 shows histograms of these distances for the original MNIST data (*left*), a RiC-VAE($N$=5,$M$=1) (*middle*), and a RiC-VAE($N$=5,$M$=5) (*right*). Clearly the RiC-VAE is not copying samples from the original data, and moreover, the additional testing passes used to generate samples for the $M$=5 case maintain these distances, while nonetheless improving the overall digit visual quality as observed previously.

## VI. CONCLUSION

Although the VAE has secured itself as a powerful generative modeling paradigm, there remain limitations to its effectiveness in practice. In this work, we have provided targeted enhancements that both reduce the sensitivity to outliers, as well as crystalize new, generated samples devoid of excessive blur. This is possible in large part due to our proposal for leveraging outputs of the generative process as virtual inputs that can be applied during training as a form of data augmentation, and during testing as a source for iterative refinements. The resulting recurrent structure itself resembles the iterative steps of certain influential compressive sensing algorithms that are also capable of incrementally removing sparse outliers. However, while the latter essentially rely on 'hand-crafted' updates derived from potentially heuristic energy function gradients or related, our pipeline is entirely learned from data.

## APPENDIX A
## PROOF OF PROPOSITION 1

Using Jensen's inequality, we have that

$$
\begin{aligned}
& 2\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})}\left[\log\left|x_j^{(i)}-\mu_{x_j}^{(i)}\right|\right] && (26)\\
\leq\ & \log \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})}\left[\left(x_j^{(i)}-\mu_{x_j}^{(i)}\right)^2\right] && (27)\\
=\ & \inf_{\lambda_j^{(i)}>0} \frac{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})}\left[\left(x_j^{(i)}-\mu_{x_j}^{(i)}\right)^2,\right]}{\lambda_j^{(i)}} + \log\lambda_j^{(i)}
\end{aligned}
$$

where an irrelevant constant has been omitted and $\boldsymbol{\lambda}^{(i)} \in \mathbb{R}_+^d$ for all $i$ represent arbitrary variational parameters, independent of $\boldsymbol{z}$. Any concave function can be expressed as a minimization of upper-bounding linear functions in this way [32]. The expectation now admits a closed-form solution leading to

$$
\begin{aligned}
& \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})}\left[\left(\boldsymbol{x}^{(i)}-\boldsymbol{W}\boldsymbol{z}-\boldsymbol{b}\right)^\top\left(\boldsymbol{\Lambda}^{(i)}\right)^{-1}\left(\boldsymbol{x}^{(i)}-\boldsymbol{W}\boldsymbol{z}-\boldsymbol{b}\right)\right]\\
=\ & \left(\boldsymbol{x}^{(i)}-\boldsymbol{W}\boldsymbol{\mu}_z^{(i)}-\boldsymbol{b}\right)^\top\left(\boldsymbol{\Lambda}^{(i)}\right)^{-1}\left(\boldsymbol{x}^{(i)}-\boldsymbol{W}\boldsymbol{\mu}_z^{(i)}-\boldsymbol{b}\right)\\
+\ & \operatorname{tr}\left[\boldsymbol{\Sigma}_z^{(i)}\boldsymbol{W}^\top\left(\boldsymbol{\Lambda}^{(i)}\right)^{-1}\boldsymbol{W}\right]. \qquad (28)
\end{aligned}
$$

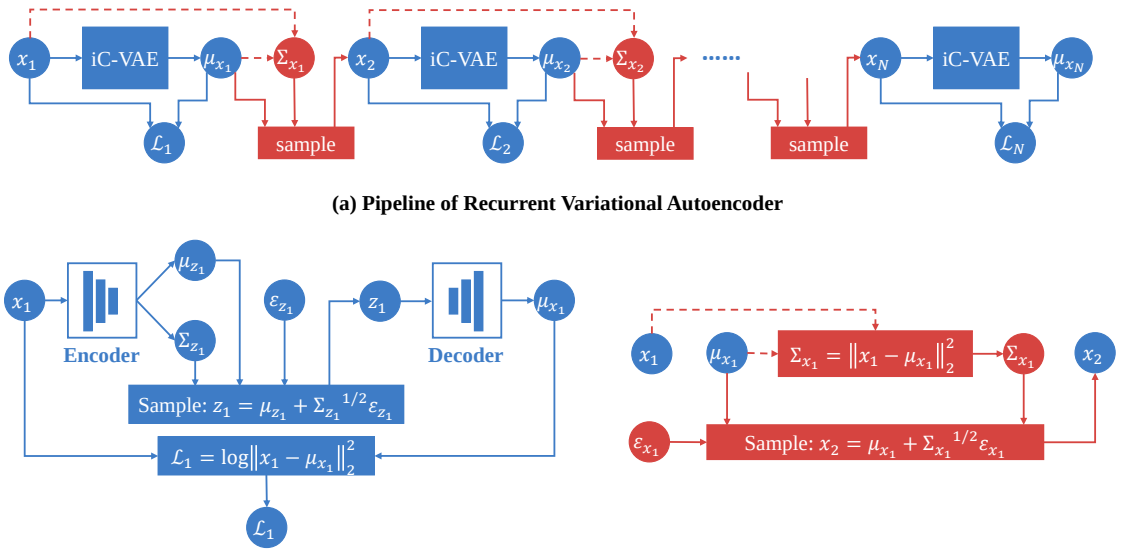
where $\boldsymbol{\Lambda}^{(i)} = \operatorname{diag}[\boldsymbol{\lambda}^{(i)}]$. Using these expressions we obtain the new upper bound on the original VAE cost given by

$$
\begin{aligned}
& \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\boldsymbol{X}) && (29)\\
\leq\ & \sum_i \inf_{\boldsymbol{\Lambda}^{(i)}\succ\mathbf{0}}\Big\{\operatorname{tr}\left[\boldsymbol{\Sigma}_z^{(i)}\right]-\log\left|\boldsymbol{\Sigma}_z^{(i)}\right|+\|\boldsymbol{\mu}_z^{(i)}\|_2^2\\
+\ & \left(\boldsymbol{x}^{(i)}-\boldsymbol{W}\boldsymbol{\mu}_z^{(i)}-\boldsymbol{b}\right)^\top\left(\boldsymbol{\Lambda}^{(i)}\right)^{-1}\left(\boldsymbol{x}^{(i)}-\boldsymbol{W}\boldsymbol{\mu}_z^{(i)}-\boldsymbol{b}\right)\\
+\ & \operatorname{tr}\left[\boldsymbol{\Sigma}_z^{(i)}\boldsymbol{W}^\top\left(\boldsymbol{\Lambda}^{(i)}\right)^{-1}\boldsymbol{W}\right]+\log\left|\boldsymbol{\Lambda}^{(i)}\right|\Big\}. \qquad (30)
\end{aligned}
$$

Because $\boldsymbol{\mu}_z^{(i)}$ and $\boldsymbol{\Sigma}_z^{(i)}$ appear in different terms, we can optimize over each separately in terms of $\boldsymbol{\Lambda}^{(i)}$ and $\boldsymbol{W}$ by taking gradients, equating to zero, and solving. This leads to the optimal solutions

$$
\begin{aligned}
\boldsymbol{\mu}_z^{(i)} &= \boldsymbol{W}^\top\left(\boldsymbol{\Lambda}^{(i)}+\boldsymbol{W}\boldsymbol{W}^\top\right)^{-1}\left(\boldsymbol{x}^{(i)}-\boldsymbol{b}\right),\\
\boldsymbol{\Sigma}_z^{(i)} &= \left[\boldsymbol{W}^\top\boldsymbol{\Lambda}^{(i)}\boldsymbol{W}+\boldsymbol{I}\right]^{-1}. \qquad (31)
\end{aligned}
$$

Plugging these values into (29) and ignoring constants, we obtain the stated bound. Moreover, we observe from this development that in fact $\boldsymbol{\mu}_z^{(i)}$ and $\boldsymbol{\Sigma}_z^{(i)}$ need not involve overly complex deep network structures. By satisfying (31), which can be viewed as simple affine




**(a) Pipeline of Recurrent Variational Autoencoder**

**(b) Details of iC-VAE**

**(c) Details of Sample Block**

Fig. 12. Detailed structure flow of the RiC-VAE network.

stationary conditions, the result still holds. ■

## APPENDIX B
## DEEP NETWORK STRUCTURE AND TRAINING DETAILS

In this section, we illustrate the further particulars of the RiC-VAE framework originally shown in Figure 2. Here Figure 12 first presents the general structure flow, including additional ingredients pertinent to the iC-VAE.

### A. RiC-VAE network structure

The dimensions of each input layer, encoder inner-product hidden layers, latent dimension of $z$, decoder inner-product hidden layers, and finally each output layer are listed here for each data set.

**Frey face (Section V-B)**
$560(input) - 1000 - 500 - 250 - 10(z) - 250 - 500 - 1000 - 560(output)$
We use ReLU activations for all inner product layers except the last one which uses sigmoid activations to accommodate the magnitude range of the image pixels.

**MNIST (Section V-A and Section V-C)**
$784(input) - 1000 - 500 - 250 - 30(z) - 250 - 500 - 1000 - 784(output)$.
The activation scheme is the same as for the Frey face data.

### B. Training Details

Learning rates and iteration counts are listed below.

**Frey face**
For iC-VAE, we set the learning rate as $1 \times 10^{-4}$ and train 150000 iterations with batch size equal to 100. For RiC-VAE($N$=2), we use the same training setting. Since the samples are actually doubled, we halve the learning rate to be $5 \times 10^{-5}$.

**MNIST**
For iC-VAE, we set the learning rate as $1 \times 10^{-4}$ and train 600000 iterations with batch size equal to 100. Then we use this to initialize the RiC-VAE($N$=5) and train an extra 100000 iterations with learning rate $2 \times 10^{-5}$. To compare iC-VAE and RiC-VAE fairly, we also train the iC-VAE for another 100000 iterations.

## REFERENCES

[1] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," in *Biological Cybernetics*, 1988.
[2] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, 2006.
[3] D. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations*, 2014.
[4] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *International Conference on Machine Learning*, 2014.
[5] B. Dai, Y. Wang, J. Aston, G. Hua, and D. Wipf, "Connections with robust pca and the role of emergent sparsity in variational autoencoder models," *Journal of Machine Learning Research*, vol. 19, no. 41, 2018.

[6] Y. Wang, B. Dai, G. Hua, J. Aston, and D. P. Wipf, "Green generative modeling: Recycling dirty data using recurrent variational autoencoders," in *Uncertainty in Artificial Intelligence*, 2017.
[7] S. Zhao, J. Song, and S. Ermon, "Infovae: Information maximizing variational autoencoders," *arXiv:1706.02262*, 2017.
[8] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, 2015.
[9] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum, "Deep convolutional inverse graphics network," in *Advances in Neural Information Processing Systems*, 2015.
[10] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," in *European Conference on Computer Vision*. Springer, 2016.
[11] C. Doersch, "Tutorial on variational autoencoders," *arXiv:1606.05908*, 2016.
[12] J. A. Palmer, D. P. Wipf, K. Kreutz-Delgado, and B. Rao, "Variational EM algorithms for non-Gaussian latent variable models," *Advances in Neural Information Processing Systems*, 2006.
[13] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis*, 2nd ed. New York: Springer-Verlag, 1985.
[14] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. Kreutz-Delgado, "Subset selection in noise based on diversity measure minimization," *IEEE Trans. Signal Processing*, vol. 51, no. 3, 2003.
[15] D. P. Wipf, "Non-convex rank minimization via an empirical Bayesian approach," in *Uncertainty in Artificial Intelligence*, 2012.
[16] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
[17] C. Y. Lee, S. Xie, and P. W. Gallagher, "Deeply supervised nets," in *Artificial Intelligence and Statistics*, 2015.
[18] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. de Freitas, "Learning to learn by gradient descent by gradient descent," in *Advances in Neural Information Processing Systems*, 2016.
[19] K. Li and J. Malik, "Learning to optimize," *arXiv:1606.01885*, 2016.
[20] P. Sprechmann, A. M. Bronstein, and G. Sapiro, "Learning efficient sparse and low rank models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, 2015.
[21] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv:1409.2574*, 2014.
[22] B. Xin, Y. Wang, W. Gao, D. Wipf, and B. Wang, "Maximal sparsity with deep networks?" in *Advances in Neural Information Processing Systems*, 2016.
[23] E. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted $\ell_1$ minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, 2008.
[24] D. P. Wipf and S. Nagarajan, "Iterative reweighted $\ell_1$ and $\ell_2$ methods for finding sparse solutions," *Journal of Selected Topics in Signal Processing (Special Issue on Compressive Sensing)*, vol. 4, no. 2, 2010.
[25] H. He, B. Xin, and D. Wipf, "From bayesian sparsity to gated recurrent nets," *arXiv:1706.02815*, 2017.
[26] W. Liu, G. Hua, and J. R. Smith, "Unsupervised one-class learning for automatic outlier removal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
[27] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, "Learning discriminative reconstructions for unsupervised outlier removal," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.
[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, 1998.
[29] E. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 2, May 2011.
[30] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, 2013.
[31] M. B. Wilk and R. Gnanadesikan, "Probability plotting methods for the analysis for the analysis of data," *Biometrika*, vol. 55, no. 1, 1968.
[32] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, no. 2, 1999.